Coding Styles Ensimag 3A SLE

Matthieu Moy

Matthieu.Moy@imag.fr

2015-2016



Matthieu Moy (Matthieu.Moy@imag.fr)

Coding Styles

2015-2016 < 1 / 28 >

- Dans la vie courante ...
- Code : fond et forme
- Directives de codage du noyau Linux

Matthieu Moy (Matthieu.Moy@imag.fr)

Coding Styles

Sommaire

2015-2016 < 2 / 28 >

Une page de journal ...



Matthieu Moy (Matthieu.Moy@imag.fr)

2015-2016

Regardons d'un peu plus près ...

pour un dirigeant mondial ou une star d'Hollywood."

L'IRA-Veritable est responsable de la plupart des attentats commis depuis 2009, et c'est l'organisation la plus meuriree. Mais, cette anée, ce sont les Oglaigh na Eireann Continuité, qui ont infligé des pertes aux forces de sceurité, quand l'explosion d'une voiture piégée a grièvement blessé un agent du PSN ne janvier. Les ONH disposent de "membres dissiminés à Derry, Dublin et dans les comis (au sud) de la frontière", selon une source. Ils sont particulièrement puissants à Armagh-Sud et surtout à Belfast, où leur commandant pratique un recrutement extrêmement agressif.

LA POLICE SUR LE TERRAIN JUGÉE IMPUISSANTE

JUGEE IMPUISSANTE
Les ONH comptent 60 membres à
Belfast, pour l'essentiel impliqués dans
des actions "punitives" (bannissement,
balle dans la rotule ou exécution
d'éléments antisociaux" tels que tra-

individus pour leur résister." En assassinant à Cork un dealer d'héroîne nooire, ITRA-Wêritable cherche aussi a conserve de l'anche de l'Arche de l'Arche

Il est personation de nois faire a magnitude contraints de nois faire a magnitude de l'ouveaux.

D'apra noi informations, le goudre de l'outrainte de l'effocerait d'entretenir le dialogue avec tous les groupes dissidents. Mais cela n'intéresse pas I'RA-Véritable. Une source proche des républicains affirme ainsi: "La lutte armée est ion d'avoir attein un niveau susceptible de pousser les Britanniauss à er retirer Par constauent, à

LES DIVIDENDE SE FONT ATTEN Dans le comté des provisoires

Matthieu Moy (Matthieu.Moy@imag.fr)

Coding Styles

2015-2016

Et encore un peu plus près ...

115 une ucciaration, la l'IRA-Véritable de Cork létenir "une liste de trafigue à exécuter", accusant gardai (police irlandaise) ınaux. Elle a désigné e potentielle un homme cié d'une remise de peine

Matthieu Moy (Matthieu.Moy@imag.fr)

2015-2016 < 6 / 28 >

Une page de journal ...

- Page découpée en plusieurs articles,
- Chaque article a un titre,
- Les articles longs sont découpés avec des sous-titres.



Matthieu Moy (Matthieu.Moy@imag.fr)

Coding Styles

 $2015\text{-}2016 \qquad < 7\,/\,28 >$

Regardons d'un peu plus près ...

- Disposition en colonnes (\approx 40 caractères de large) ⇒ lecture plus facile (retour des yeux à la ligne suivante).
- Découpage du texte en paragraphe ⇒ Saut de paragraphe = petite pause, passage à une autre notion.



Et encore un peu plus près ...

 Placement des espaces et ponctuation

 $\overset{\cdot}{\Rightarrow}$ lecture fluide.

- ► Pas d'espaces à l'intérieur des parenthèses
- Espaces autour des parenthèses
- ► Espace après une virgule mais pas avant

l'IRA-Véritable de Cork létenir "une liste de trafigue à exécuter", accusant gardai (police irlandaise) ınaux. Elle a désigné potentielle un homme zié d'une remise de peine

Matthieu Moy (Matthieu.Moy@imag.fr) Coding Styles 2015-2016 < 8 / 28 > Matthieu Moy (Matthieu.Moy@imag.fr) Coding Styles 2015-2016

Et si on faisait autrement?

Tout informaticien doit connaître le langage C.C.' est une espèce d'espéranto de l'informatique. Les autres langages fournissent en effet souvent une interface avec C(ce qui leur permet en particulier de s'interfacer plus facilement avec le système d'exploitation) ou sont eux-mêmes écrits en C.D'autre part c'est le langage de base pour programmer les couches basses des systèmes informatiques. Par exemple, on écrit rarement un pilote de périphérique en Ada ou Java. Finalement en compilation, C est souvent choisi comme cible de langages de plus haut niveau. Toutefois, il est peu probable(ou plus exactement, peu souhaitable) qu'un ingénieur informaticien soit confronté à de gros développements logiciels entièrement en C.L'objectif pédagogique du projet est donc surtout de montrer comment C peut servir d'interface entre les langages de haut niveau et les couches basses de la machine.Plus précisément, les objectifs du stage C de première année sont : Apprentissage de C (en soi, et pour la démarche qui consiste à apprendre un nouveau langage). Lien du logiciel avec les couches basses de l'informatique, ici logiciel un peu conséquent, à développer dans les règles de l 'art(mise en œuvre de tests, documentation, démonstration du logiciel, partage du travail, ...) Lien avec les autres modules de première année (théorie des langages) et de deuxième année (compilation, système)

Matthieu Moy (Matthieu.Moy@imag.fr)

Coding Styles

2015-2016 < 10 / 28 >

Tentons de faire mieux ...

Tout informaticien doit connaître le langage C. C'est une espèce d'espéranto de l'informatique. Les autres langages fournissent en effet souvent une interface avec C (ce qui leur permet en particulier de s'interfacer plus facilement avec le système d'exploitation) ou sont eux-mêmes écrits en C. D'autre part c'est le langage de base pour programmer les couches basses des systèmes informatiques. Par exemple, on écrit rarement un pilote de périphérique en Ada ou Java. Finalement, en compilation, C est souvent choisi comme cible de langages de plus haut

Toutefois, il est peu probable (ou plus exactement, peu souhaitable) qu'un ingénieur informaticien soit confronté à de gros développements logiciels entièrement en C. L'objectif pédagogique du projet est donc surtout de montrer comment C peut servir d'interface entre les langages de haut niveau et les couches basses de la machine. Plus précisément, les objectifs du stage C de première année sont:

première année sont :
Apprentissage de C (en soi, et pour la démarche qui consiste à apprendre un nouveau langage).
Lien du logiciel avec les couches basses de l'informatique, ici logiciel de base et architecture.
Le premier projet logiciel un peu conséquent, à développer dans les règles de l'art (mise en œuvre de tests, documentation, démonstration du logiciel, partage du travail, ...)
Lien avec les autres modules de première année (théorie des langages) et de deuxième année (compilation, système)

Matthieu Moy (Matthieu.Moy@imag.fr)

Coding Styles

2015-2016

< 12 / 28 >

Tentons de faire mieux ...

Tout informaticien doit connaître le langage C. C'est une espèce d'espéranto de l'informatique. Les autres langages fournissent en effet souvent une interface avec C (ce qui leur permet en particulier de s'interfacer plus facilement avec le système d'exploitation) ou sont eux-mêmes écrits en C. D'autre part c'est le langage de base pour programmer les couches basses des systèmes informatiques. Par exemple, on écrit rarement un pilote de périphérique en Ada ou Java. Finalement, en compilation, C est souvent choisi comme cible de langages de plus haut niveau.

Toutefois, il est peu probable (ou plus exactement, peu souhaitable) qu'un ingénieur informaticien soit confronté à de gros développements logiciels entièrement en C. L'objectif pédagogique du projet est donc surfout de montrer comment C peut servir d'interface entre les langages de haut niveau et les couches basses de la machine. Plus précisément, les objectifs du stage C de première année sont :

- Apprentissage de C (en soi, et pour la démarche qui consiste à apprendre un nouveau legrane)
- Lien du logiciel avec les couches basses de l'informatique, ici logiciel de base et architecture
- Le premier projet logiciel un peu conséquent, à développer dans les règles de l'art (mise en œuvre de tests, documentation, démonstration du logiciel, partage du travail, ...)
- Lien avec les autres modules de première année (théorie des langages) et de deuxième année (compilation, système)

Matthieu Moy (Matthieu.Moy@imag.fr) Coding Styles 2015-2016 < 14 / 28 >

Principe de la « fenêtre brisée »

- Une fenêtre brisée
- ⇒ le vent s'engouffre, des saletés entrent
- « mon voisin n'entretient pas ses fenêtres, pourquoi je le ferais? »



- Une ligne mal indentée
- ⇒ « je ne vais pas corriger, ça va faire des conflits dans Git »
- $\Rightarrow\,$ « pourquoi je m'embêterait à formatter mon code proprement ? »
- $\Rightarrow\,$ « pourquoi j'écrirais de la doc, de toutes façons, le code est sale »
- ⇒ « la dernière fois que j'ai codé, je n'ai pas écrit de tests, ça s'est bien passé quand même ... »

Tentons de faire mieux ...

Tout informaticien doit connaître le langage C. C'est une espèce d'espéranto de l'informatique. Les autres langages fournissent en effet souvent une interface avec C (ce qui leur permet en particulier de s'interfacer plus facilement avec le système d'exploitation) ou sont eux-mêmes écrits en C. D'autre part c'est le langage de base pour programmer les couches basses des systèmes informatiques. Par exemple, on écrit rarement un pilote de périphérique en Ada ou Java. Finalement, en compilation, C est souvent choisi comme cible de langages de plus haut niveau. Toutefois, il est peu probable (ou plus exactement, peu souhaitable) qu'un ingénieur informaticien soit confronté à de gros développements logiciels entièrement en C. L'objectif pédagogique du projet est donc surtout de montrer comment C peut servir d'interface entre les langages de haut niveau et les couches basses de la machine. Plus précisément, les objectifs du stage C de première année sont : Apprentissage de C (en soi, et pour la démarche qui consiste à apprendre un nouveau langage). Lien du logiciel avec les couches basses de l'informatique, ici logiciel de base et architecture. Le premier projet logiciel un peu conséquent, à développer dans les règles de l'art (mise en œuvre de tests, documentation, démonstration du logiciel, partage du travail, ...) Lien avec les autres modules de première année (théorie des langages) et de deuxième année (compilation, système)

Matthieu Moy (Matthieu.Moy@imag.fr)

Coding Styles

2015-2016

< 11 / 28 >

Tentons de faire mieux ...

Tout informaticien doit connaître le langage C. C'est une espèce d'espéranto de l'informatique. Les autres langages fournissent en effet souvent une interface avec C (ce qui leur permet en particulier de s'interfacer plus facilement avec le système d'exploitation) ou sont eux-mêmes écrits en C. D'autre part c'est le langage de base pour programmer les couches basses des systèmes informatiques. Par exemple, on écrit rarement un pilote de périphérique en Ada ou Java. Finalement, en compilation, C est souvent choisi comme cible de langages de plus haut niveau.

inveau. Toutefois, il est peu probable (ou plus exactement, peu souhaitable) qu'un ingénieur informaticien soit confronté à de gros développements logiciels entièrement en C. L'objectif pédagogique du projet est donc surtout de montrer comment C peut servir d'interface entre les langages de haut niveau et les couches basses de la machine. Plus précisément, les objectifs du stage C de première année sont:

- Apprentissage de C (en soi, et pour la démarche qui consiste à apprendre un nouveau langage).
- Lien du logiciel avec les couches basses de l'informatique, ici logiciel de base et architecture.
- Le premier projet logiciel un peu conséquent, à développer dans les règles de l'art (mise en œuvre de tests, documentation, démonstration du logiciel, partage du travail, ...)
- Lien avec les autres modules de première année (théorie des langages) et de deuxième année (compilation, système)

Matthieu Moy (Matthieu.Moy@imag.fr)

Coding Styles

2015-2016

< 13 / 28 >

Conclusion

- Le fond est important
- ... mais on a du mal à se concentrer sur le fond si la forme est imparfaite!

→ C'est pareil pour du code!

Matthieu Moy (Matthieu.Moy@imag.fr)

Matthieu Moy (Matthieu.Moy@imag.fr)

Coding Styles

2015-2016 < 15 / 28 >

2015-2016

< 18 / 28 >

Peut-on faire mieux?

Coding Styles

```
struct noeud_t{
unsigned val;
struct noeud_t *fg;
struct noeud_t * fd;
};

int rechercher(unsigned v,struct noeud_t *arbre)
{
   int res;
   if(arbre != NULL) {
       if (v==l= NULL) {
        return 1;
       } else if(v(arbre->val) {
        return rechercher(v,arbre->fg);
       }
       else
       {
        res =rechercher( v, arbre -> fd ) ;
        return res;
       }
   } else {
        return 0;
   }
}
```

Matthieu Moy (Matthieu Moy@imag.fr) Coding Styles 2015-2016 < 16 / 28 >

Peut-on faire mieux? struct noeud_t{ unsigned val; struct noeud_t *fg; struct noeud_t * fd; int rechercher(unsigned v,struct noeud_t *arbre) int res; if(arbre != NULL) { if (v==arbre->val) { return 1; } else if(v<arbre->val){ return rechercher(v,arbre->fg); res =rechercher(v, arbre -> fd) ; return res ; return 0 ;

Coding Styles

2015-2016

< 19 / 28 >

Matthieu Moy (Matthieu.Moy@imag.fr)

```
Peut-on faire mieux?
struct noeud t {
           unsigned val;
struct noeud_t *fg;
struct noeud_t *fd;
};
int rechercher(unsigned v, struct noeud_t *arbre)
           if (arbre == NULL) {
    return 0;
           } else {
                      if (v == arbre->val) {
                      return 1;
} else if (v < arbre->val) {
    return rechercher(v, arbre->fg);
                      } else {
                                 res = rechercher(v, arbre->fd);
return res;
                                                                                         < 21 / 28 >
Matthieu Moy (Matthieu.Moy@imag.fr)
                                            Coding Styles
                                                                              2015-2016
```

```
Peut-on faire mieux?
struct noeud t {
         unsigned val;
         struct noeud_t *fg;
         struct noeud t *fd;
};
int rechercher (unsigned v, struct noeud t *arbre)
         if (arbre == NULL) {
         return 0;
} else if (v == arbre->val) {
         return 1;
} else if (v < arbre->val) {
         return rechercher(v, arbre->fg);
else {
                  return rechercher(v, arbre->fd);

→ on va s'arrêter là!

                                                             2015-2016 < 23 / 28 >
Matthieu Moy (Matthieu.Moy@imag.fr)
                                   Coding Styles
```

Indentation, présentation du code

- Indentation à 8 caractères : blocs if/while/for/... clairement visibles.
- Pas de ligne de plus de 80 caractère : plus lisible, affichable sur une petite fenêtre.
- Une fonction doit tenir sur un écran (en théorie, 1 écran = 25 lignes). Faire des fonctions courtes, qui font une chose et qui le font bien.
- Si on a l'impression qu'il y a conflit entre les deux premiers points, c'est qu'il faut mieux découper (cf. point 3).
- Utiliser les lignes vides pour marquer une pause (cf. notion de paragraphe en français)

Peut-on faire mieux?

```
struct noeud_t {
           unsigned val;
struct noeud_t *fg;
struct noeud_t *fd;
int rechercher(unsigned v, struct noeud_t *arbre)
           int res;
           if (arbre != NULL) {
                     return 1;
} else if (v < arbre->val) {
  return rechercher(v, arbre->fg);
} else {
                                 res = rechercher(v, arbre->fd);
                                 return res ;
                      return 0;
```

Peut-on faire mieux?

Coding Styles

```
struct noeud_t {
           unsigned val:
           struct noeud_t *fg;
struct noeud_t *fd;
};
int rechercher(unsigned v, struct noeud_t *arbre)
           if (arbre == NULL) {
                      return 0;
           } else {
                     . GIDIE->val) {
   return 1;
} else if (v < arbre->val) {
   return rechercher(v, arbre->fg);
} else {
   ....
                      if (v == arbre->val) {
                                 return rechercher(v, arbre->fd);
}
```

Matthieu Moy (Matthieu.Moy@imag.fr)

Matthieu Moy (Matthieu.Moy@imag.fr)

Coding Styles

2015-2016 < 22 / 28 N

2015-2016 < 20 / 28 >

Directives de codage du noyau Linux

- Utilisées par Linux et beaucoup d'autres projets
- Dispo un peu partout sur le Web (http://lxr.linux.no/#linux/Documentation/CodingStyle)
- Bon ou mauvais style : question subjective, mais l'important est d'être homogènes!

"First off, I'd suggest printing out a copy of the GNU coding standards, and NOT read it. Burn them, it's a great symbolic gesture."

Matthieu Moy (Matthieu.Moy@imag.fr) Coding Styles

2015-2016 < 25 / 28 >

Nommage des fonctions, variables

- Noms courts et expressifs
 - ▶ ThisVariableIsATemporaryCounter \Rightarrow trop long
 - ▶ a, var1 ⇒ pas expressifs
- Nom expressif indispensable pour les variables/fonctions globales.

Matthieu Moy (Matthieu.Moy@imag.fr) Coding Styles 2015-2016 < 26 / 28 > Matthieu Moy (Matthieu.Moy@imag.fr) Coding Styles 2015-2016 < 27 / 28 >

Commentaires	
Expliquez pourquoi votre code est comme il est, et non comment.	
 Si le code a besoin de beaucoup de commentaire pour expliquer comment il fonctionne, c'est qu'il est trop complexe ⇒ simplifiez-le 	
plutôt que de le commenter!	
Matthieu Moy (Matthieu.Moy@imag.fr) Coding Styles 2015-2016 < 28 / 28 >	